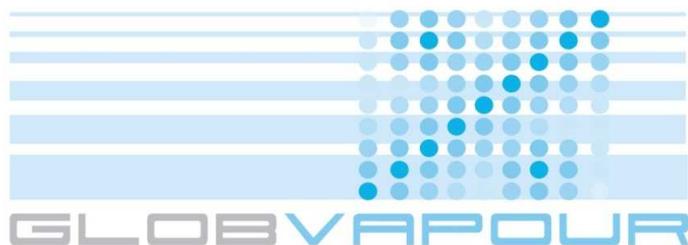# DUE GLOBVAPOUR

Processing System
Validation Report

Issue 1 Revision 0

19 January 2012

Project nr: ESRIN/AO/1-6090/09/I-OL

Project Coordinator: Marc Schröder
                       Deutscher Wetterdienst
                       marc.schroeder@dwd.de

Document Change Record

Document, Version	Date	Changes	Originator
0.1	16.07.2011	Template extracted from validation plan	Bö
0.9	18.01.2012	Factory test run performed and documented Output verification section and confirmations by partners added	OF Bö
1.0	19.01.2012	Version 1.0 of Processing System Validation Report (SVR)	Nadine Schneider

Table of Contents

1   Summary

This test report records the observations and results of the factory acceptance test run performed at Brockmann Consult. The test run uses the test cases and procedures defined in the GlobVapour Processing System Validation Plan (version 0.9). This chapter identifies the test cases as repetition from the test plan.

1.1 Test Cases Overview

Table 1-1 lists the test cases for the GlobVapour stand-alone processing system. Each test case is described by an identifier GV-TC-<number>, a title, and the system requirements from the GlobVapour Processing System Design Document (version 0.8). to be verified by the test case (forward tracing). The inverse trace from requirements to test cases is listed in the test plan.

Table 1-1: GlobVapour stand-alone processing system test cases

ID	Test Case Title	System Requirements
GV-TC 10	SSM/I 1DVAR processor run	GV-SR-50, GV-SR-100
GV-TC 20	SSM/I level 3 processor run	GV-SR-50, GV-SR-100
GV-TC 30	MERIS level 2 processor run	GV-SR-60
GV-TC 40	MERIS level 3 processor run	GV-SR-60
GV-TC 50	SSM/I MERIS blending processor run	GV-SR-60
GV-TC 60	SSM/I MERIS chain production	GV-SR-50, GV-SR-60, GV-SR-90, GV-SR-110, GV-SR-120, GV-SR-150, GV-SR-260, GV-SR-270
GV-TC 70	GOME chain production	GV-SR-70, GV-SR-90, GV-SR-110, GV-SR-130
GV-TC 80	AATSR production	GV-SR-80, GV-SR-90, GV-SR-110, GV-SR-140
GV-TC 90	Interruption and resume	GV-SR-290, GV-SR-300
GV-TC 100	Processing failure handling	GV-SR-280, GV-SR-300
GV-TC 110	Input ingestion	GV-SR-190
GV-TC 120	Result retrieval	GV-SR-200
GV-TC 130	Partial cleanup	GV-SR-210

Each of the test cases is defined with its inputs, expected results and pass/fail criteria in the following subsections.

In addition to this other requirements can be verified by inspection during the test. See the test plan for a list of requirements and their verification method.

1.2 Main Result

Main results are

- The test is passed with all test cases successfully executed.

- A few tests had observations that have been analysed

- The comparison of outputs of the stand-alone processing system with operational outputs has been performed with equality for SSM/I and GOME and with explained minor differences for MERIS.

Main issue during the test run was that the "expected" outputs had been generated with older versions of processors such that the numerical differences lead to large diff outputs, which is our method to verify that the outputs are equal. This is mitigated by the confirmation of consistency of the stand-alone processing system outputs by the respective partners as listed in the third chapter.

That the update of processor versions during and after the pre-computation of the results has lead to a situation where it is not the newest version of the MERIS processor that has generated all results. But we are convinced that

- the new version of the MERIS processor is functionally correct, as demonstrated by the tests. It has generated every type of output successfully

- the new version of the MERIS processor shows better performance

- FUB has verified the outputs of the old processor 1.3.002 and has used the analysis of differences to the operational products to improve the processor, which results in the version 1.4.030

So, we propose to include the new FUB 1.4.030 in the processing system.

2 Test Procedures and Protocol

This chapter lists the instructions performed for the tests for the GlobVapour stand-alone processing system defined in the previous chapter and the protocol by the test engineer. Each test ends with a disposition that is summarised in the last subsection.

Version record		
Steps	Notes	Rerun
Record the date of the test run.	17.01.2012	18.01.2012
Record the names of the test engineer and test observers.	O. Faber (OF)	OF

Determine the versions of processors used:
 	
`cd inst`
`cat mygv`
`cd`
 	
This prints out paths to processors. Add the versions listed in the notes column in the following steps.
	ok	ok

Record the instance version.		
Record the SSMI 1dvar processor version.	2.00	2.00
Record the SSMI l2tol3 processor version.	2.01	2.01
Record the MERIS FUB version of all MERIS processors.	1.4.030	1.4.030
Record the GOME processor version in the notes.	3.0	3.01
Record the BEAM version in the notes.	4.9	4.9

Note by the testers: The test has been performed using the partially pre-computed results for 2007-07 obtained with earlier versions of processors for MERIS (1.3.002) and GOME (0.9). This is necessary to allow to perform and repeat the test in reasonable time. At the time of pre-computing the processing mode used was "I" for SSM/I and "F" for MERIS. To harmonise at least the input and output modes and file names for this test run and use end mode "E" in all steps we have re-named all products in the archive from _I_ or _F_ to _E_ manually.

2.1 Test Procedures for Processor Runs

GV-TC-10 SSM/I 1DVAR processor run		
Steps	Notes	Rerun
To run the SSM/I 1DVAR processor:		

Steps	Notes	Rerun
```cd inst . mygv cd ~/cache mkdir test-10 cd test-10 ssmi-l2.sh \ /home/gvop/eodata/SSMI_L1C/v1/2007/07/01/cmsaf.cbtf.ssmi.f13.2007-07-01.l1c.nc \ /home/gvop/cache/test-10/2007/07``` This will write out messages on stdout initially and will run for some time. You may monitor CPU usage for the tools "cdo", "collocate.x" and "SSMIS_SAFProg.o" with ```top``` in another terminal.	total time 23'	total time 24'
To check the return code ```echo $? # expected value: 0``` To check availability and size of the output file: ```find 2007/07 -ls # expected size: 75797000```	ok	ok
To check the content ```ncdump 2007/07/SSMI*nc > output.cdl ncdump \ ~/eodata/SSMI_L2/v1/2007/07/SSMIxxxxxxx_L2_xx_F13_20070701000000_*.nc \ > expected.cdl diff expected.cdl output.cdl``` Compare with expected output. Only a single line should differ: ```1c1 < netcdf SSMIxxxxxxx_L2_xx_F13_20070701000000_E_20111229203718 { --- > netcdf SSMIxxxxxxx_L2_xx_F13_20070701000000_E_20120109103017 {```	ok	(test repeated to verify the _E_ in the output) ok
Delete cache ```cd ~/cache rm -r test-10```	done	done
Enter disposition (passed/to be repeated/failed)	passed	passed

GV-TC-20 SSM/I level 3 processor run		
Steps	Notes	Rerun
To run the SSM/I l2tol3 processor: ```cd inst . mygv cd ~/cache mkdir test-20 cd test-20 ssmi-l3.sh /home/gvop/eodata/SSMI_L2/v1/2007/07/*nc \ ../test-20-output/2007/07```	total time 20'	total time 25'

Steps	Notes	Rerun
This will write out messages on stdout and will run for some time. You may monitor CPU usage for the tools "l2l3_ssmi" with `top` in another terminal.		

To check the return code
 	
`echo $?`
`# expected value: 0`
 	
To check availability and size of the output files:
 	
`ls -l ../test-20-output/2007/07/`
`# expected output size of 31 DC files: 4413364`
`# expected output size of 1 MM file: 5450392`
	ok	ok

To check the content
 	
`ncdump \`
`../test-20-output/2007/07/SSMIxxxxxxx_L3_MM_Fxx_20070701000000_E_2012??????????.nc \`
`> mm.cdl`
`ncdump \`
`~/eodata/SSMI_L3/v1/2007/07/SSMIxxxxxxx_L3_MM_Fxx_20070701000000_E_20111230073554.nc \`
`> mm-expected.cdl`
`diff mm-expected.cdl mm.cdl`
 	
Compare with expected output. Only a few lines should differ. Example:
 	
`1c1`
`< netcdf SSMIxxxxxxx_L3_MM_Fxx_20070701000000_E_20111230073554 {`
`---`
`> netcdf SSMIxxxxxxx_L3_MM_Fxx_20070701000000_E_20120109120009 {`
`53c53`
`<          :filename = "SSMIxxxxxxx_L3_MM_Fxx_20070701000000_E_20111230073554.nc" ;`
`---`
`>          :filename = "SSMIxxxxxxx_L3_MM_Fxx_20070701000000_E_20120109120009.nc" ;`
`55,56c55,56`
`<          :timestamp = "20111230073554" ;`
`<          :history = "created on 2011-12-30 07:35:54 UTC (4199)" ;`
`---`
`>          :timestamp = "20120109120009" ;`
`>          :history = "created on 2012-01-09 12:00:09 UTC (4199)" ;`
	the difference is very large	ok (unclear why there was a difference in the first run)

Delete cache
 	
`cd ~/cache`
`rm -r test-20`
	done	done

Enter disposition (passed/to be repeated/failed)	to be repeated	passed

GV-TC-30 MERIS level 2 processor run		
Steps	Notes	Rerun

To run the MERIS level 2 processor:
 	
`cd inst`
`. mygv`
`cd ~/cache`
`mkdir test-30`
`cd test-30`
`mkdir -p dim/2007/07/01`		

Steps	Notes	Rerun

```beam-pconvert.sh \```
`/home/gvop/eodata/MER_RR__1P/r03/2007/07/01/MER_RR__1PRACR20070701_000250`
`_000026412059_00274_27883_0000.N1 \`
```dim/2007/07/01```
```meris-l2.sh \```
`` `pwd`/dim/2007/07/01/MER_RR__1PRACR20070701_000250_000026412059_00274_27883_0000.dim \``
```l2/2007/07/01```
 	
This will write out messages on stdout and will run for some time.
You may monitor CPU usage for the tools "java pconvert" and "python gv_meris_l2_processing.py" with
```top```
in another terminal.
	total time 8'
	total time 22'
		total time 7'

To check the return code
```echo $?```
```# expected value: 0```
To check availability and size of the output files:
```ls -l l2/2007/07/01```
```# expected output size: 5450392```
	ok
	
	obtained output size: 286028608
		(expected value is wrong)

To check the content
```ncdump l2/2007/07/01/MERIS*.nc > output.cdl```
```ncdump \```
`~/eodata/MERIS_L2/v1/2007/07/01/MERISxxxxx-L2-xx-xxx-20070701000250-E-20111230024449.nc \`
```> expected.cdl```
```diff expected.cdl output.cdl```
Compare with expected output. Only a few lines should differ. Example:
`1c1`
`< netcdf MERISxxxxx-L2-xx-xxx-20070701000250-E-20111230024449 {`
`---`
`> netcdf MERISxxxxx-L2-xx-xxx-20070701000250-E-20120109173538 {`
`44c44`
`<          :filename = "output/2007/07/01/MERISxxxxx-L2-xx-xxx-20070701000250-E-20111230024449.nc" ;`
`---`
`>          :filename = "output/2007/07/01/MERISxxxxx-L2-xx-xxx-20070701000250-E-20120109173538.nc" ;`
`51c51`
`<          :timestamp = "20111230024449" ;`
`---`
`>          :timestamp = "20120109173538" ;`
`59c59`
`<          :history = "created on 2011-12-30 02:44:49 UTC" ;`
`---`
`>          :history = "created on 2012-01-09 17:35:38 UTC" ;`
	the difference is very large	(Diff. explained by FUB version difference 1.3 (expected) and 1.4 (output))

Delete cache
```cd ~/cache```
```rm -r test-30```
		done

Enter disposition (passed/to be repeated/failed)	check	passed

GV-TC-40 MERIS level 3 processor run		
Steps	Notes	Rerun
To run the MERIS level 3 processor: ```cd inst```		

Steps	Notes	Rerun

```. mygv```
```cd ~/cache```
```mkdir test-40```
```cd test-40```
```meris-dc.sh \```
```/home/gvop/eodata/MERIS_L2/v1/2007/07/01/MER_RR__2PRACR20070701_*.N1 \```
```dc/2007/07/01```
 	
This will write out messages on stdout and will run for some time.
You may monitor CPU usage for the tool "python gv_meris_dc_processing.py" with
```top```
in another terminal.
	total time < 1'	

To check the return code
```echo $?```
```# expected value: 0```
To check availability and size of the output files:
```ls -l dc/2007/07```
```# expected output size: 388845844```
	ok	

To check the content
```ncdump dc/2007/07/01/MERIS*.nc > output.cdl```
```ncdump \```
```~/eodata/MERIS_L3/v1/2007/07/MERISxxxxx_L3_DC_ENV_20070701120000_E_20111230032358.nc \```
```> expected.cdl```
```diff expected.cdl output.cdl```
Compare with expected output. Only a few lines should differ. Example:
```1c1```
```< netcdf MERISxxxxx_L3_DC_ENV_20070701120000_E_20111230032358 {```
```---```
```> netcdf MERISxxxxx_L3_DC_ENV_20070701120000_E_20120109163829 {```
```55c55```
```<   :filename = "output/2007/07/MERISxxxxx_L3_DC_ENV_20070701120000_E_20111230032358.nc" ;```
```---```
```>   :filename = "output/2007/07/MERISxxxxx_L3_DC_ENV_20070701120000_E_20120109163829.nc" ;```
```62c62```
```<   :timestamp = "20111230032358" ;```
```---```
```>   :timestamp = "20120109163829" ;```
```70c70```
```<   :history = "created on 2011-12-30 03:23:58 UTC" ;```
```---```
```>   :history = "created on 2012-01-09 16:38:30 UTC" ;```
	the difference is very large	(Diff. explained by FUB version difference 1.3 (expected) and 1.4 (output))

Delete cache
```cd ~/cache```
```rm -r test-40```
		done

Enter disposition (passed/to be repeated/failed)	check	passed

GV-TC-50 SSMI-MERIS blending processor run		
Steps	**Notes**	**Rerun**

To run the SSMI MERIS blending processor:
```cd inst```
```. mygv```
```cd ~/cache```
```mkdir test-50```		

	total time 2'	

```cd test-50```
```ssmi-meris-dc.sh \```
```/home/gvop/eodata/MERIS_L3/v1/2007/07/MERISxxxxx_L3_DC_ENV_2007070112000\```
```0_E_20111230032358.nc \```
```/home/gvop/eodata/SSMI_L3/v1/2007/07/SSMIxxxxxxx_L3_DC_Fxx_200707*nc \```
```dc/2007/07```
 	
This will write out messages on stdout and will run for some time.
You may monitor CPU usage for the tool "python gv_merge_meris_ssmi_processing.py" with
```   top```
in another terminal.
	total time 2'	

To check the return code
```echo $?```
```# expected value: 0```
To check availability and size of the output files:
```ls -l dc/2007/07```
```# expected output size: 492526332```
	obtained output size: 492526340	(Explained by version difference between UCAS 0.9 and 3.0)

To check the content
```ncdump dc/2007/07/*.nc > output.cdl```
```ncdump \```
```~/eodata/SSMI_MERIS_L3/v1/2007/07/SSMI_MERIS_L3_DC_xxx_20070701120000_E_20111230104307.nc \```
```> expected.cdl```
```diff expected.cdl output.cdl```
Compare with expected output. Only a few lines should differ. Example:
```1c1```
```< netcdf SSMI_MERIS_L3_DC_xxx_20070701120000_E_20111230104307 {```
```---```
```> netcdf SSMI_MERIS_L3_DC_xxx_20070701120000_E_20120109170220 {```
```61c61```
```<   :filename   =```
```"output/2007/07/SSMI_MERIS_L3_DC_xxx_20070701120000_E_20111230104307.nc" ;```
```---```
```>   :filename   =```
```"output/2007/07/SSMI_MERIS_L3_DC_xxx_20070701120000_E_20120109170220.nc" ;```
```68c68```
```<   :timestamp = "20111230104307" ;```
```---```
```>   :timestamp = "20120109170220" ;```
```76c76```
```<   :history = "created on 2011-12-30 10:43:07 UTC" ;```
```---```
```>   :history = "created on 2012-01-09 17:02:21 UTC" ;```
	the difference is very large	(Explained by version difference between UCAS 0.9 and 3.0)

Delete cache
```cd ~/cache```
```rm -r test-50```
		done

Enter disposition (passed/to be repeated/failed)	check	passed

2.2 Test Procedures for Production with Processing Chains

GV-TC-60 SSM/I MERIS chain production		
Steps	**Notes**	**Rerun**
To remove some results from the archive directory tree: ```cd test/backuptestoutputs.sh find eodata/backup -type f	wc -l``` The number of files moved away is expected to be 267.	ok	
To prepare the instance to a state where these outputs are still to be generated: ```cd mv test/gv.report.test inst/gv.report ls -l inst``` The size of the gv.report file is expected to be 375607 bytes.	ok	

To start the processing system:
 	
``cd inst
. mygv
gvstartup```
 	
This command is expected to start the processing system and write out one summary status line and four lines for concurrent processes being started:
 	
``1034 created, 4 running, 80 backlog, 950 processed, 0 failed
r /home/gvop/inst/bin/gome-l3.sh /home/gvop/eodata/GOME2_L2/v1/2007/07/H2O_2007_07_01.global....
r /home/gvop/inst/bin/ssmi-l2.sh /home/gvop/eodata/SSMI_L1C/v1/2007/07/31/cmsaf.cbtf.ssmi.f13....
r /home/gvop/inst/bin/ssmi-l2.sh /home/gvop/eodata/SSMI_L1C/v1/2007/07/31/cmsaf.cbtf.ssmi.f14....
r /home/gvop/inst/bin/beam-pconvert.sh /home/gvop/eodata/MER_RR__1P/r03/2007/07/31/MER_RR__1P...```
 	
The system will run for several hours to complete processing.
	ok
	
	total time about 4 hours	

To monitor the processing system, use any of the following. To list the running tasks (r as first letter in line) and maybe failures (f as first letter inline):
 	
```cat gv.status```
or in a separate window:
```watch -n 10 cat gv.status```
To list the completed tasks (in a separate window):
```tail -f gv.report```
To list operating system processes:
```gvps|grep inst|grep -v bin/sh|grep -v grep|grep -v gvps```
To list CPU, memory and I/O usage (in a separate window):
```top```
The load is not completely balanced over the time. Some tasks are I/O intensive and will not use all CPUs. Others will.
	ok	(The system had been restarted after a failure of the host machine of the VM after 3 hours)

	Notes	Rerun

Check termination by the summary line of gv.status
 	
`..., 0 running, ...`
 	
Double check by using
 	
`ps auxww|grep gv.py`
 	
which is expected NOT to list a process.
 	
Check success by gv.status that is expected to contain only the summary line:
 	
`1034 created, 0 running, 0 backlog, 1034 processed, 0 failed`
 	
In case of failures the tasks failed are listed. Reasons can be found in the most recent log files:
 	
`ls -ltr log/`
	ok	

The last line of gv.report is expected to contain the path to the SSM/I MERIS MM composite for the test month:
 	
`#output /home/gvop/eodata/SSMI_MERIS_L3/v1/2007/07 /home/gvop/eodata/SSMI`
`_MERIS_L3/v1/2007/07/SSMI_MERIS_L3_MM_xxx_20070701120000_E_....nc`
 	
Check the content
 	
`cd ~/cache`
`mkdir test-60`
`cd test-60`
`ncdump  /home/gvop/eodata/SSMI_MERIS_L3/v1/2007/07/SSMI_MERIS_L3_MM_*.nc`
`\`
`> output.cdl`
`ncdump \`
`/home/gvop/eodata/backup/SSMI_MERIS_L3/v1/2007/07/SSMI_MERIS_L3_MM_*.nc \`
`> expected.cdl`
`diff expected.cdl output.cdl`
 	
Compare with expected output. Only a few lines should differ. Example:
 	
`1c1`
`< netcdf SSMI_MERIS_L3_MM_xxx_20070701120000_E_20111230104307 {`
`---`
`> netcdf SSMI_MERIS_L3_MM_xxx_20070701120000_E_20120109170220 {`
`61c61`
`<            :filename                                         =`
`"output/2007/07/SSMI_MERIS_L3_MM_xxx_20070701120000_E_20111230104307.nc" ;`
`---`
`>            :filename                                         =`
`"output/2007/07/SSMI_MERIS_L3_MM_xxx_20070701120000_E_20120109170220.nc" ;`
`68c68`
`<            :timestamp = "20111230104307" ;`
`---`
`>            :timestamp = "20120109170220" ;`
`76c76`
`<            :history = "created on 2011-12-30 10:43:07 UTC" ;`
`---`
`>            :history = "created on 2012-01-09 17:02:21 UTC" ;`
	the difference is very large	(Diff. explained by FUB version difference 1.3 (expected) and 1.4 (output))

Delete cache
 	
`cd ~/cache`
`rm -r test-60`
		done

Enter disposition (passed/to be repeated/failed)	check	passed

GV-TC-70 GOME chain production		
Steps	Notes	Rerun
GOME processing is performed already in GV-TC-60 as one of the first processing tasks. What remains is the verification of results.		

To check availability and size of the output file:
<pre>ls -l ~/eodata/GOME_L3/v1/2007/
# expected output size: 4153904</pre> 	
	obtained output size 4413404	(Explained by version difference between UCAS 0.9 and 3.0)

To check the content
<pre>cd ~/cache
mkdir test-70
cd test-70
ncdump ~/eodata/GOME2_L3/v1/2007/GOME2.nc > output.cdl
ncdump ~/eodata/backup/GOME_L3/v1/2007/GOME2*nc \
> expected.cdl
diff expected.cdl output.cdl</pre>Compare with expected output. Only a few lines should differ. Example:
<pre>1c1
< netcdf GOME2xxxxxx_L3_MC_M02_20070700000000_E_20111229213421 {
--- 	
> netcdf GOME2xxxxxx_L3_MC_M02_20070700000000_E_20120110102858 {
42c42
<   :filename = "GOME2xxxxxx_L3_MC_M02_20070700000000_E_20111229213421.nc" ;
--- 	
>   :filename = "GOME2xxxxxx_L3_MC_M02_20070700000000_E_20120110102858.nc" ;
44,45c44,45
<   :timestamp = "20111229203507" ;
<   :history = "2011-12-29 20:35:09 UTC: merge L3 GDP 4.4" ;
--- 	
>   :timestamp = "20120110093000" ;
>   :history = "2012-01-10 09:30:01 UTC: merge L3 GDP 4.4" ;</pre>
	the differ-ence is very large	(Explained by version difference between UCAS 0.9 and 3.0)

Delete cache
<pre>cd ~/cache
rm -r test-70</pre>
		done

Enter disposition (passed/to be repeated/failed)	check	passed

2.3 Test Procedures for Contingency Cases

GV-TC-90 Interruption and resume		
Steps	Notes	Rerun

To remove some results from the archive directory tree:
<pre>cd
mv eodata/backup eodata/backup1
test/backuptestoutputs.sh
find eodata/backup -type f | wc -l</pre>The number of files moved away depends on the final status of GV-TC-60.
	ok	

To prepare the instance to a state where some outputs are still to be generated: ``` cd mv test/gv.report.test inst/gv.report ls -l inst ``` The size of the gv.report file is expected to be 375607 bytes.	ok	
To reduce the concurrency of the system in order to simplify the test case edit `inst/gv.py`. Change ``` do_ssmi = True do_meris = True do_gome = True ``` to ``` do_ssmi = False do_meris = True do_gome = False ``` and change ``` hosts = [('localhost',19)] ``` to ``` hosts = [('localhost',5)] ```	ok	
To start the processing system: ``` cd inst . mygv gvstartup ``` This command is expected to start the processing system and write out one summary status line and one line for a process being started: ``` 332 created, 1 running, 287 backlog, 44 processed, 0 failed r /home/gvop/inst/bin/beam-pconvert.sh /home/gvop/eodata/MER_RR__1P/r03/2007/07/31/MER_RR__1PRACR20070731_010141_000026342060_00203_28313 _0000.N1 /home/gvop/eodata/MER_RR__1P/dim/2007/07/31 ```	ok	
To interrupt processing, stop the monitor and if necessary computing processes: ``` gvshutdown ps auxww|egrep 'java|gv.py|cdo|collocate.x| SSMIS_SAFProg.out|l2l3_ssmi' kill <pid> # where <pid> is the listed process of the processor ps auxww|egrep 'java|gv.py|cdo|collocate.x| SSMIS_SAFProg.out|l2l3_ssmi' ``` The verification is expected to list no process of a processor nore the python process executing gv.py.	ok	
Resume processing at that state: ``` gvstartup cat gv.status ``` The expected output is that the interrupted task is started again: ``` 312 created, 1 running, 268 backlog, 43 processed, 0 failed r /home/gvop/inst/bin/beam-pconvert.sh /home/gvop/eodata/MER_RR__1P/r03/2007/07/31/MER_RR__1PRACR20070731_010141_000026342060_00203_28313 _0000.N1 /home/gvop/eodata/MER_RR__1P/dim/2007/07/31 ```	ok	
Interrupt processing again to use the same system status for the next test case.	done	
Enter disposition (passed/to be repeated/failed)	passed	

GV-TC-100 Processing failure handling		
Steps	**Notes**	**Rerun**
Ensure that the initial status is the same as for GV-TC-90.		

Replace the input of the next step by a corrupted file:
<pre>cd
mkdir eodata/backup2
mv \
/home/gvop/eodata/MER_RR__1P/r03/2007/07/31/MER_RR__1PRACR20070731_010141_000026342060_00203_28313_0000.N1 \
eodata/backup2
touch \
/home/gvop/eodata/MER_RR__1P/r03/2007/07/31/MER_RR__1PRACR20070731_010141_000026342060_00203_28313_0000.N1
mv \
/home/gvop/eodata/MER_RR__1P/dim/2007/07/31/MER_RR__1PRACR20070731_010141_000026342060_00203_28313_0000* \
~/eodata/backup2/
ls -l /home/gvop/eodata/MER_RR__1P/r03/2007/07/31</pre>The size of the corrupted input file is expected to be 0 bytes.
	ok	

To start the processing system:
<pre>cd inst
. mygv
gvstartup</pre>This command is expected to start the processing system and write out one summary status line and one line for a process being started:
<pre>332 created, 1 running, 287 backlog, 44 processed, 0 failed
r /home/gvop/inst/bin/beam-pconvert.sh
/home/gvop/eodata/MER_RR__1P/r03/2007/07/31/MER_RR__1PRACR20070731_010141_000026342060_00203_28313_0
000.N1 /home/gvop/eodata/MER_RR__1P/dim/2007/07/31</pre>
	ok	

Verify the status after some seconds:
<pre>head -n3 gv.status</pre>The expected output is that the task with the corrupted input has failed:
<pre>326 created, 0 running, 281 backlog, 44 processed, 1 failed
f /home/gvop/inst/bin/beam-pconvert.sh
/home/gvop/eodata/MER_RR__1P/r03/2007/07/31/MER_RR__1PRACR20070731_010141_000026342060_00203_28313_0
000.N1 /home/gvop/eodata/MER_RR__1P/dim/2007/07/31
...</pre>
	ok	

Interrupt processing again to use the same system status for the next test case.	done	

Replace the input of the next step by the correct input file again:
<pre>cd
mv -f \
eodata/backup2/MER_RR__1PRACR20070731_010141_000026342060_00203_28313_0000.N1 \
/home/gvop/eodata/MER_RR__1P/r03/2007/07/31
ls -l /home/gvop/eodata/MER_RR__1P/r03/2007/07/31</pre>The size of the first input file is expected to be larger than 0 bytes.
	ok	

Resume processing at that state:
<pre>cd inst
gvstartup
sleep 3
cat gv.status</pre>The expected output is that the interrupted task is started again:		

```312 created, 1 running, 268 backlog, 43 processed, 0 failed```
```r /home/gvop/inst/bin/beam-pconvert.sh```
```/home/gvop/eodata/MER_RR__1P/r03/2007/07/31/MER_RR__1PRACR20070731_010141_000026342060_00203_28313_0```
```000.N1 /home/gvop/eodata/MER_RR__1P/dim/2007/07/31```
	ok	

Enter disposition (passed/to be repeated/failed)	passed	

2.4 Test Procedures for Data Management

GV-TC-110 Input ingestion		
Steps	**Notes**	**Rerun**

Look up additional SSM/I input data:
 	
```cd```
``find eodata/additionalinputs -type f
	less```
	``find eodata/additionalinputs -type f
		wc -l```
		
		15 files are expected to be listed.
			ok	

Add the input data to the archive:
 	
```rsync -rv eodata/additionalinputs/MER_RR__1P/r03/2007/08 \```
```eodata/MER_RR__1P/r03/2007```
 	
Verify that the data has been copied:
 	
``find eodata/MER_RR__1P/r03/2007/08 -type f
	less```
	``find eodata/MER_RR__1P/r03/2007/08 -type f
		wc -l```
		
		The number of files is expected to be the same as in the first step.
			ok	

Enter disposition (passed/to be repeated/failed)	passed	

GV-TC-120 Result retrieval		
Steps	**Notes**	**Rerun**

Look up result data:
 	
```cd```
``find eodata/SSMI_MERIS_L3/v1/2007/07 -type f
	less```
	```du -s eodata/SSMI_MERIS_L3/v1/2007/07```
		eodata/ SSMI_ MERIS _L3/v1/ 2007/0 7 is empty	(TC must be done before conting ency cases)

Add the input data to the archive:
 	
```mkdir -p eodata/gv-export/SSMI_MERIS_L3/v1/2007```
```rsync -rv eodata/SSMI_MERIS_L3/v1/2007/07 \```
```eodata/gv-export/SSMI_MERIS_L3/v1/2007```
 	
Verify that the data has been copied:
```du -sh eodata/gv-export/```
	ok after copy SSMI_	

```# expected 7.4G```
 ```du -sh eodata/SSMI_MERIS_L3/v1/2007/07/```
 ```# expected 7.4G```
 The size of the source directory is expected to be in the same order as the result directory.
	MERIS _L3 data from backup

Enter disposition (passed/to be repeated/failed)	passed

GV-TC-130 Partial cleanup		
Steps	Notes	Rerun

Determine archive space and use:
 ```cd```
 ```df eodata/```
 ```du -s eodata/*```
	ok	

Remove the additional input data from the archive:
 ```cd```
 ```mkdir -p eodata/trash```
 ```mv eodata/MER_RR__1P/r03/2007/08 eodata/trash```
 ```rm -r eodata/trash```
 Verify that the data has been removed:
 ```du -s eodata/*```
 The size of the MER_RR__1P directory is expected to be smaller as in the first step.
	Obs.: rm command had to be repeated	

Enter disposition (passed/to be repeated/failed)	passed	

2.5 Test Summary

Test result	
Steps	Notes
Number of tests (finally) passed:	12 of 12
Number of tests failed:	0
Enter disposition (passed, or updates required, tests to be repeated)	passed

3   Comparison of Test Outputs with Operational Products

3.1 GOME2 Level 3 MM

Two cycles of verification and version updates of the UCAS processor has been performed before we have reached the status of identity of outputs. Reasons where a newer UCAS version at DLR and an internal configuration parameter. Both issues have been fixed. The stand-alone processing system now contains the actual UCAS version 3.01.

The monthly mean for 2007-07 processed with the stand-alone processing system and UCAS version 3.01 has been submitted to DLR for verification 18.01.2012 .

B. Aberle of DLR has verified the output and has confirmed that the product is identical to the DLR output in an email 18.01.2012 :

```
-------- Original-Nachricht --------
Betreff: Re: [globvapour] ucas_delivery_V3.0.tgz - test in stand-alone processing system
Datum: Wed, 18 Jan 2012 18:16:05 +0100
Von: Bernd Aberle <bernd.aberle@dlr.de>
An: Martin Boettcher <martin.boettcher@brockmann-consult.de>
Kopie (CC): 'bc@globvapour.info' <bc@globvapour.info>, Schröder Marc <Marc.Schroeder@dwd.de>,
Schneider Nadine <Nadine.Schneider@dwd.de>, Diego Loyola <Diego.Loyola@dlr.de>

Hallo Martin,

We generated water vapour products in NetCDF format on
different platforms: the development platform at DLR and the
production platform at Brockmann Consult using the current
UCAS version 3.01 and identical input files.
The generated outputs files turned out to be identical as well
and show basicly reasonable water vapour maps with the expected
higher water contentration in tropical regions and lower
concentration over desert and mountain areas.

Gruss,
Bernd


> Am 2012-01-18 10:15, schrieb Martin Boettcher:
> Dear Bernd,
>
> please find attached the ucas 3.01 output for 2007-07 processed with the stand-alone
processing system. If it is correct, please confirm this for the test report. Note that the
result differs from the file you have provided with the processor.
>
> Best regards,
> Martin and Olga
```

3.2 SSM/I DC and MM

No cycle has been required to reach equality of results for SSM/I.

Two SSM/I L2, a DC and MM product processed with the stand-alone processing system with the GlobVapour SSM/I 1DVAR processor 2.00 and the L2-to-L3 processor 2.01 have been submitted to DWD for verification 09.01.2012 .

Nadine Schneider of DWD has verified the intermediates and outputs and has confirmed that the products are identical to the ones produced by DWD in an email at 09.01.2012 :

```
-------- Original-Nachricht --------
Betreff: [globvapour] SSMI Verification of outputs of the stand-alone processing system
Datum: Mon, 9 Jan 2012 16:50:22 +0000
Von: Schneider Nadine <Nadine.Schneider@dwd.de>
An: Schneider Nadine <Nadine.Schneider@dwd.de>, 'Martin Boettcher' <martin.boettcher@brockmann-
consult.de>, "'rasmus.lindstrot@wew.fu-berlin.de'" <rasmus.lindstrot@wew.fu-berlin.de>,
"'bernd.aberle@dlr.de via F*EX service fex.dlr.de'" <bernd.aberle@dlr.de>
Kopie (CC): 'Schröder Marc' <Marc.Schroeder@dwd.de>, 'Rene Preusker' <rene.preusker@wew.fu-
berlin.de>, 'Diego Loyola' <Diego.Loyola@dlr.de>

Dear Martin, dear GlobVapour-Team,

I hereby verify, that the following output of the GV stand-alone processing system is identical
to the output of the DWD processing system.

SSMIxxxxxxx_L2_xx_F13_20070701000000_I_20111229203718.nc
SSMIxxxxxxx_L2_xx_F14_20070701000000_I_20111229233333.nc
SSMIxxxxxxx_L3_DC_Fxx_20070701000000_I_20111230073554.nc
SSMIxxxxxxx_L3_MM_Fxx_20070701000000_I_20111230073554.nc

Best regards, Nadine


-----Ursprüngliche Nachricht-----
Von: Martin Boettcher [mailto:martin.boettcher@brockmann-consult.de]
Gesendet: Montag, 9. Januar 2012 09:24
An: Schneider Nadine; 'rasmus.lindstrot@wew.fu-berlin.de'; bernd.aberle@dlr.de via F*EX service
fex.dlr.de
Cc: Schröder Marc; Rene Preusker; Diego Loyola
Betreff: [globvapour] Verification of outputs of the stand-alone processing system

Dear Nadine, dear Rasmus, dear Bernd,

we wish you a happy new year!

Last year we have produced the output products for one month 07/2007 with the GlobVapour stand-
alone processing system. May I ask you to verify that these outputs are correct for 'your'
respective sensor? We would like to add your response to the test protocol for the dry run.

The products are:

SSMIxxxxxxx_L2_xx_F13_20070701000000_I_20111229203718.nc
SSMIxxxxxxx_L2_xx_F14_20070701000000_I_20111229233333.nc
SSMIxxxxxxx_L3_DC_Fxx_20070701000000_I_20111230073554.nc
SSMIxxxxxxx_L3_MM_Fxx_20070701000000_I_20111230073554.nc
MERISxxxxx-L2-xx-xxx-20070701000250-F-20111230024449.nc
MERISxxxxx_L3_DC_ENV_20070701120000_F_20111230032358.nc
MERISxxxxx_L3_MM_ENV_20070701120000_F_20111230033955.nc
SSMI_MERIS_L3_DC_xxx_20070701120000_E_20111230104307.nc
SSMI_MERIS_L3_MM_xxx_20070701120000_E_20111230122549.nc
GOME2xxxxxx_L3_MC_M02_20070700000000_I_20111229213421.nc

They are available as a package at

ftp://ftp-cmsaf.dwd.de/incoming/GlobVapour/gv-standalone-output-verification-package.tar.gz

They have been generated by:

ssmi 1dvar 2.01
ssmi-l2tol3 2.00
meris-fub 1.3-002
ucas 0.9

Thanks!
Best regards
Martin
```

3.3 MERIS DC and MM and SSM/I-MERIS DC and MM

The verification of MERIS outputs and merged SSM/I-MERIS outputs has been done concurrently with performance improvements of the processor implementation. The production of the dataset has required several weeks processing time. For this reason, the outputs compared have been those generated with FUB 1.3-002. Two update cycles with intermediate version FUB 1.4-025 have been performed with the following changes since then:

- configuration of processing mode (E for End mode, I for Internal mode, ...) for input and output file names
- smaller memory footprint to allow concurrent processes
- algorithmic improvements

Though the verification of results has been done on the FUB 1.3-002 outputs we have decided to test and deliver the stand-alone processing system with the actual FUB 1.4-030. Reasons for this:

- The performance of the system is better, it does no longer require 8 GB main memory.
- R. Preusker has confirmed that the numerical differences between the two versions are small, and the results of FUB 1.4-030 must be considered principally better

Note that ss a consequence, the outputs of the stand-alone processing system test run are mixed version outputs for MERIS. Most of the L2 and L3 DC processing has been done with FUB 1.3-002 while some L2, one L3 DC and the MM as well as all merged SSM/I-MERIS DC and MM has been generated with FUB 1.4-030.

The mail of R. Preusker, FUB, confirms that

- the outputs of the MERIS L2 processing is identical to the FUB results. Minor deviations between the original IDL version outputs and the FUB 1.3-002 python outputs are fixed in FUB 1.4-030.
- the outputs of MERIS L3 processing are identical (except for differences caused by L2 differences)
- the merged SSM/I-MERIS composites show issues at the coast line. They are improved in FUB 1.4-030.

```
-------- Original-Nachricht --------
Betreff: globvapor FUB test
Datum: Tue, 17 Jan 2012 01:46:58 +0100
Von: Rene Preusker <rene.preusker@gmx.de>
Antwort an: rene.preusker@wew.fu-berlin.de
Organisation: FU-Berlin
An: Martin Boettcher <martin.boettcher@brockmann-consult.de>
Kopie (CC): Rasmus Lindstrot <rasmus.lindstrot@googlemail.com>

Hallo Martin,

wir haben die Ergebnisse des L2 getestet:
1. Unser python  L2 Prozessor lieferte die gleichen Ergebnisse, wie
   der von euch implementierte. Ihr habt also richtig implementiert.
2. Es gab seltsame kleine Abweichungen über Australien
   zwischen der ursprünglichen idl Version und meiner,
   die ich vorher nicht bemerkt hatte. Diese Unterschiede
   könnte man  "selten und gering aber kritisch" nennen.
   Die Ursache war ein falsches DEM, wenn  der Satelliten-
   track in mehr als einen  Abschnitt unterteilt wurde.
   Der Fehler liegt also auch in der  Version,
   die ich euch als letztes geschickt habe.
   Eine fehlerbereinigte Version ist hier:
   ftp://160.45.77.9//pub/gv_meris_FUB_1.4.030.tar.gz
```

```
3. Es gibt weitere kleine Unterschiede, die aber nur
   irrelevante Auswirkungen haben und deshalb unkritisch sind:
   + die Küsten werden etwas anders behandelt
   + die Wolkenränder / Umgebungen  werden etwas anders definiert
   + die LUTs werden leicht unterschiedlich vorbereitet
   + die auxilary snow/icemask  werden leicht unterschiedlich benutzt


3. Die MERIS L3 Produkte stimmen überein
   (bis auf die kleinen Differenzen im L2 durch das falsche  DEM)

4. Im Python L3 Prozessor laufen die sun-glint Werte mit,
   im idl Prozessor noch nicht.
   Dadurch sieht man in den  MERIS L3  über einen
   Breitengradbereich  im Ozean  schrägen Streifen.
   Das ist so gewollt und richtig.

5. Das gemergte Meris-SSMI Produkt hat noch diese hässlichen Ränder,
   an der Küste. Ich dachte, dass ich sie  ausgemerzt hatte.
   Es geht mittlerweile noch besser (mit der o.g. Version)

Ich weiss nicht, ob ihr nocheinmal einen Monat prozessieren könnt,
der L2 prozessor ist jetzt ja deutlich schneller... Aber zumindest das
Ssmi-Meris-Mergen sollte man nochmal machen, damit die Küstenränder
verschwinden.

Viele Grüße

   Rene

--
_____
Dr. Rene Preusker

Institut für Weltraumwissenschaften
Freie Universität Berlin
Carl-Heinrich-Becker-Weg 6-10
D-12165 Berlin

Tel.: +49 30 838 52375
Fax.: +49 30 838 56664
e-mail: rene.preusker@wew.fu-berlin.de
_____
```